# An Empirical Study for Defect Prediction using Clustering

[1]Ms. Puneet Jai Kaur and [2]Ms. Pallavi
[1]Assistant Professor, UIET, Panjab University, Chandigarh.
puneetkaur79@yahoo.co.in
[2]UIET, Panjab University, Chandigarh.
pallavigoyal19@yahoo.in

***Abstract: -*** **Reliably predicting defects in the software is one of the holy grails of software engineering. Researchers have devised and implemented a method of defect prediction approaches varying in terms of accuracy, complexity, and the input data they require. An accurate prediction of the number of defects in a software product during system testing contributes not only to the management of the system testing process but also to the estimation of the product's required maintenance [1]. A prediction of the number of remaining defects in an inspected artefact can be used for decision making. Defective software modules cause software failures, increase development and maintenance costs, and decrease customer satisfaction. It strives to improve software quality and testing efficiency by constructing predictive models from code attributes to enable a timely identification of fault-prone modules [2]. In this paper, we will discuss clustering techniques are used for software defect prediction. This helps the developers to detect software defects and correct them. Unsupervised techniques may be used for defect prediction in software modules, more so in those cases where defect labels are not available [3].**

***Keywords:*** **data mining, defect prediction, hierarchical clustering, k-mean clustering, density-based clustering.**

## I. Introduction

Software engineering discipline contains several prediction approaches such as test effort prediction, reusability prediction, correction cost prediction, fault prediction, security prediction, effort prediction and quality prediction. Software fault prediction is most popular research area in these prediction approaches. Software defect prediction approaches use previous fault data to predict fault-prone modules for the next release of software. The success of the software system depends not only on cost and schedule but also on quality. The prediction result, which is the number of defects remaining in a software system, can be used as an important measure for the software developer, and can be used to control the software process and gauge the likely delivered quality of a software system. In this paper, we will discuss about how clustering techniques are used for software defect prediction. Clustering involves finding natural groupings in data. Unsupervised learning methods such as clustering techniques are a natural choice for analyzing software quality in the absence of fault proneness labels. Clustering algorithms can group the software modules according to the values of their software metrics. The underlying software engineering assumption is that fault-prone software modules will have similar software measurements and so will likely form clusters. Similarly, not-fault-prone modules will likely group together. When the clustering analysis is complete, a software engineering expert inspects each cluster and labels it *fault prone* or *not fault prone*. A clustering approach offers practical benefits to the expert who must decide the labels. Instead of inspecting and labelling software modules one at a time, the expert can inspect and label a given cluster as a whole; he or she can assign all the modules in the cluster the same quality label.

K-means algorithm is widely used for clustering because of its computational efficiency. K-means seeks a set of k cluster centres so as to minimize the sum of the squared Euclidean distance between each point and its nearest cluster centre. K-means starts with a set Z of centres and computes their neighbourhoods. In each iteration, every centre is moved to the centroid of its neighbourhood and then the neighbourhoods are recomputed based on the updated positions of the k centres. This process continues until a convergence criterion is satisfied; for instance, a given number of iterations have been performed or successive iterations produce no changes to any of the k neighbourhoods. The collection of neighbourhoods that results is taken to be the partition of the data points produced by k-means applied to the initial set of centres.

Hierarchical clustering builds a cluster hierarchy i.e. a tree of clusters. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. But a series of partitions takes place, which may vary from a single cluster containing all objects to n clusters each, containing a single object.

There are two main methods of hierarchical clustering algorithm are agglomerative or divisive.

First method is agglomerative approach, where we start from the bottom where all the objects are going up (bottom up approach) through merging of objects. We begin with each individual objects and merge the two closest objects. The process is iterated until all objects are aggregated into a single group [5].

Second method is divisive approach (top down approach), where we start with assumption that all objects are group into a single group and then we split the group into two recursively until each group consists of a single object. One possible way to perform divisive approach is to first form a


DOI: 03.LSCS.2013.4.582

minimum spanning tree (e.g. using Kruskal algorithm) and then recursively (or iteratively) split the tree by the largest distance [8].

In a density based clustering a cluster is defined as maximal set of density connected points. Clusters are identified by looking at the density of points. Regions with a high density of points depict the existence of clusters whereas regions with a low density of points indicate clusters of noise or clusters of outliers. This algorithm is particularly suited to deal with large datasets, with noise, and is able to identify clusters with different sizes and shapes.

The rest of the paper is organized as follows: Section 2 presents clustering techniques for Software Defect Prediction. Section 3 presents the conclusion and future work.

## II. Clustering Techniques for software defect prediction

In this paper, different clustering techniques are discussed for identifying fault prone modules. Clustering plays an important role in software defect prediction. Following are the clustering techniques:

### *A. K-means clustering*

*K*-means is a partitioned clustering technique that is well-known and widely used for its low computational cost. They often produce clusters of relatively uniform sizes, even if input data have varied cluster sizes, which are called the uniform effect. The *k*-means algorithms perform iteratively the partition step and new cluster centre generation step until convergence [4]. The clustering result guarantees a local minimum solution only. These algorithms are very sensitive to the initial cluster centres. For simplicity, users often use the random initialization method to obtain an initial set of cluster centres. However, these clustering algorithms need to rerun many times with different initializations in an attempt to find an optimal solution [6].

### *B. Hierarchical clustering*

We presented a fault prediction model using hierarchical clustering to estimate the software quality. In order to achieve a high quality development faults must be known prior to development so that more and smart emphasis can put in to that particular areas. Hierarchical clustering solutions which are in the form of trees called *dendrograms* are of great interest for a number of application domains. Hierarchical trees provide a view of the data at different levels of abstraction [7]. The consistency of clustering solutions at different levels of granularity allows flat partitions of different granularity to be extracted during data analysis, making them ideal for interactive exploration and visualization. In addition, there are many times when clusters have sub clusters, and hierarchical structures represent the underlying application domain naturally. Hierarchical clustering solutions have been primarily obtained using agglomerative algorithms in which objects are initially assigned to their own cluster and then pairs of clusters are repeatedly merged until the whole tree is formed. However, partitioned algorithms can also be used to obtain hierarchical clustering solutions via a sequence of repeated bisections [8], [9].

### *C. Density-Based Clustering*

There is lot of work done in prediction of the faults and fault proneness in the various kinds of software systems. But, it is the impact or level of severity of those faults which is more important than number of faults existing in the systems, as the major faults matters most for a developer than the minor ones and these major faults needs immediate attention. Density-Based Spatial Clustering of Applications is most widely used density based algorithm and has played a significant role in finding non linear shapes structure based on the density in various application domains. In density based clustering, Clusters are identified by looking at the density of points. Regions with a high density of points depict the existence of clusters whereas regions with a low density of points indicate clusters of noise or clusters of outliers. In a density based clustering a cluster is defined as maximal set of density connected points. The main feature of density based clustering is that it discovers features of arbitrary shape and it can handle noise. Deduce the results on basis of accuracy, precision and recall values [10].

## III. Conclusion and Future Work

In this paper, we have discussed about all the clustering techniques that can be used for software defect prediction. K-mean clustering is the most common technique that is used for software defect prediction, but we will be working on hierarchical divisive clustering for defect prediction. We will compare the result of the hierarchical clustering with the k-mean clustering to find which is better for software defect prediction.

## References

[1] Lourdes Pelayo and Scott Dick, "Evaluating Stratification Alternatives to Improve Software Defect Prediction", IEEE TRANSACTIONS ON RELIABILITY, VOL. 61, NO. 2, JUNE 2012.

[2] Qinbao Song, Martin Shepperd, Michelle Cartwright, and Carolyn Mair, "Software Defect Association Mining and Defect Correction Effort Prediction, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 2, FEBRUARY 2006

[3] Shi Zhong, Taghi M. Khoshgoftaar, and Naeem Seliya, "Analyzing Software Measurement Data with Clustering Techniques" Published by the IEEE Computer Society in 2004.

[4] Michael Laszlo and Sumitra Mukherjee, "A Genetic Algorithm Using Hyper-Quad trees for Low-Dimensional K-means Clustering", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 4, APRIL 2006.

[5] Arshdeep Kaur and Sunil Gulati, "A Framework for Analyzing Software Quality using Hierarchical Clustering", International Journal on Computer Science and Engineering, Vol. 3, No. 2 Feb 2011.

[6] Jiye Liang, Liang Bai, Chuangyin Dang, "The K-Means-Type Algorithms Versus Imbalanced Data Distributions", IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 20, NO. 4, AUGUST 2012.



DOI: 03.LSCS.2013.4.582

[7] YING ZHAO, GEORGE KARYPIS, "Hierarchical Clustering Algorithms for Document Datasets", 2005 Springer Science.

[8] http://people.revoledu.com/kardi/tutorial/Clustering/Hierarchical%20Clustering.

[9] Jayanthi Ranjan and Dr. S.I Ahson, "Efficient Agglomerative Method for Micro Array Data on breast cancer Outcome", International Conference on Cognitive Systems, December 2004.

[10] Parvinder S. Sandhu, Sheena Singh and Neha Budhija, "Prediction of Level of Severity of Faults in Software Systems using Density Based Clustering", 2011 International Conference on Software and Computer Applications IPCSIT vol.9, 2011.


DOI: 03.LSCS.2013.4.582